# How to Make Personalized Web Browsing Simple, Secure, and Anonymous

Eran Gabber, Phillip B. Gibbons, Yossi Matias, Alain Mayer

Bell Laboratories, Lucent Technologies
600 Mountain Avenue, Murray Hill NJ 07974 USA
{eran,gibbons,matias,alain}@research.bell-labs.com

**Abstract.** An increasing number of web-sites require users to establish an account before they can access the information stored on that site ("personalized web browsing"). Typically, the user is required to provide at least a unique username, a secret password and an e-mail address. Establishing accounts at multiple web-sites is a tedious task. A security- and privacy-aware user may have to invent a distinct username and a secure password, both unrelated to his/her identity, for each web-site. The user may also desire mechanisms for anonymous e-mail. Besides the information that the user supplies voluntarily to the web-site, additional information about the user may flow (involuntarily) from the user's site to the web-site, due to the nature of the HTTP protocol and the cookie mechanism.

This paper describes the Janus Personalized Web Anonymizer, which makes personalized web browsing simple, secure and anonymous by providing convenient solutions to each of the above problems. Janus serves as an intermediary entity between a user and a web-site. Given a user and a web-site, Janus automatically generates an alias – typically a username, a password and an e-mail address – that can be used to establish an anonymous account at the web-site. Different aliases are generated for each user, web-site pair; however the same alias is presented whenever a particular user visits a particular web-site. Janus frees the user from the burden of inventing and memorizing distinct usernames and secure passwords for each web-site, and guarantees that an alias (including an e-mail address) does not reveal the true identity of the user. Janus also provides mechanisms to complete an anonymous e-mail exchange from a web-site to a user, and filters the information-flow of the HTTP protocol to preserve user privacy. Thus Janus provides simultaneous *user identification* and *user privacy*, as required for *anonymous personalized web browsing*.

## 1 Introduction

The Internet has become an important place for businesses to offer information and services to potential customers. Popular examples of such businesses are news providers (e.g., `www.nytimes.com`, `www.hotwired.com`, `www.zdnet.com`), book stores (e.g., `www.amazon.com`), car manufacturers (e.g., `www.ford.com/us`, `www.gm.com`, `www.toyota.com`), and many more. Typically, a company sets up a web-site for that purpose, and often includes its electronic address in its regular advertisements. Potential customers use web-browsers (e.g., Netscape Navigator, Microsoft Explorer) to access the information offered on these sites.

An increasing number of sites offer **personalized service**. A common example is providing personalized web pages, such that a user visiting the site is presented with hyper-links and displayed messages tailored according to the user's preferences. Such preferences can be ascertained by requiring a user to establish an account with that site, and login to his/her account on each subsequent visit. The site learns of a user's preferences either by tracking the hyper-links the user followed or through explicit dialogs with the user. The site associates this information with the user's account, and uses it to provide personalized web pages to returning users. For example, Yahoo (`my.yahoo.com`) and Ziff Davies (`www.zdnet.com`) provide "personalized news" to each user, where the sequence and selection of information is customized according to preferences associated with the user's account. In order to open an account at sites providing personalized service, the user typically has to provide a username, a password and an e-mail address. The latter is often used by the web-site to send information that is not provided on the site itself.

Establishing accounts at multiple web-sites is a tedious task for a user. Each web-site requires a new username to be distinct from all previous usernames. Thus, individuals with common names will more than likely be unable to use simple combinations of their first or last names as usernames, and instead would have to come up with and remember a more complicated username. Moreover, reusing the same username for many accounts would enable a coalition of web-sites to learn a user's browsing habits and build a profile (dossier) on the user. Thus a user who values his/her privacy will need to invent and remember a potentially large number of usernames, such that each username is unrelated to his/her identity. The more accounts the user has and the more people are connected to the Internet, the more tedious this task will become. The problem is even more severe for passwords, since good passwords are more difficult to invent and remember. Thus users are left with unsatisfactory choices, as echoed by Dave Taylor in the Usenix newsletter *;login:* "I use the same account name and password for all. That is not very bright and presents some potential security problems, but it's a survival mechanism" [T96].

Making matters worse, there are commercial products available that allow web-sites to track their clients and visitors. Such tracking can be done even when no voluntary information is provided by the user (e.g., no account is established). One such system is "SiteTrack" (`www.cortex.net/sitetrack`), whose advertisement reads as follows: "Identify who is visiting your site; Record the actual number of people that visit (not plain hits); Find which links they follow and trace their complete path; Learn which site users came from and which site they depart to..." These products are made possible by the fact that the HTTP protocol (see [HTTP]), on which the World Wide Web is based, allows specific information to flow back from the user to the web-site. This can include the previous web-site visited by the user, as well as information about the user's software and host-computer.

Finally, certain sites (e.g., the NEW YORK TIMES site) require the user to provide a valid e-mail address in order to establish an account. Providing the same e-mail address to each such site enables a coalition of web-sites to build a dossier on the user; moreover, the e-mail address itself can often reveal much about the user's identity.

### 1.1 The Janus Personalized Web Anonymizer

From the above discussion, we conclude that there is a simultaneous need for the seemingly conflicting objectives of *user identification* (so that users can sign up for personalized services) and *user privacy* in web browsing. We term this combined goal *anonymous personalized web browsing*. This paper describes the Janus[1] Personalized Web Anonymizer, the first system (of which we are aware) to provide users with a convenient means for anonymous personalized web browsing. Janus serves as an intermediary entity (a proxy) between a user and a web-site. It achieves the goals of user identification and user privacy by automatically generating aliases for users; such aliases allow the user to login to his/her accounts using a pseudonym that hides the user's true identity. An alias is typically a username, a password and an e-mail address. Different aliases are generated for each user, web-site pair; however the same alias is presented whenever a particular user visits a particular web-site. Janus frees the user from the burden of inventing and memorizing unique (alias-) usernames and secure passwords for each web-site. Moreover, the user no longer has to type in such usernames and passwords every time he/she returns to a web-site requiring an account; instead Janus will provide the appropriate usernames and passwords automatically.

The user provides a uniquely identifying e-mail address and a secret to Janus once at the start of a browsing session. Janus will use this input to generate all aliases for the user during that session. The secret can be considered as the user's "universal password" for all of the user's web-site accounts. The secret does not have to be globally unique over all users of Janus (which would be hard to enforce). For further privacy, Janus does not maintain any information about a user (such as his/her secret) who is not currently in a browsing session.

Janus also provides alias e-mail addresses. These are of the form "alias-email@hostname", where "hostname" is a trusted intermediary machine which is part of the Janus system, and "alias-email" is a string that hides the identity of the user and enables the intermediary (but no web-site) to compute the user's real e-mail address. Janus provides mechanisms to complete an anonymous e-mail exchange from a web-site to a user.

In addition, Janus filters the data stream from a user's browser to web-sites in order to ensure that no privacy-compromising information is passed on (without the user's explicit approval). Note that a web-site may not be aware that a user is accessing it via a Janus proxy.

[1] Janus is the Roman god with two faces.

At the core of anonymous personalized web browsing is the problem of *names translation*: translating from the user's e-mail address and secret to an alias that fulfills a number of properties, including anonymity, consistency, secrecy, uniqueness of an alias, and protection from creation of dossiers. To address these requirements, we use a cryptographic function that we define in a companion paper [BGGMM97], to provide for any interaction that is personalized, yet anonymous. This function is called a *Janus function.* The cryptographic framework and tools developed in [BGGMM97] are only briefly sketched in this paper. More generally, this framework may be used in other contexts in which users interact with services, and where it is desired to maintain both personalization and anonymity. Furthermore, [BGGMM97] introduces a more advanced scheme (with a more complex implementation) for integrating alias e-mail; this scheme reduces the required trust in the intermediary.

Finally, note that there are two aspects to anonymity in web browsing: anonymity of data content and anonymity of connection. The former means that the content of any data flowing between users and web-sites obeying the HTTP protocol does not reveal the user's identity; the latter means that the web-site (and eavesdroppers and other intruders) cannot identify the user via other aspects of the connection (e.g., by timing analysis). Janus guarantees anonymity of data content; anonymity of connection has been studied in contexts such as anonymous electronic cash and e-mail (see [GWB97] for an overview of such work), and we assume these previous techniques can be used to provide anonymity of connection in web browsing.

The Janus Personalized Web Anonymizer is currently being implemented. Further details can be found in the Janus home page, `http://www.bell-labs.com/project/janus`.

*Outline of this Paper.* Section 2 presents the architecture of the Janus system. Section 3 sketches the Janus function and shows its use within our system. In Section 4, we show how to incorporate e-mail sent by web-sites into the Janus system. Section 5 briefly considers options to use Janus together with an electronic payment system. Section 6 discusses related work and Section 7 concludes.

## 2 Janus Overview

In this section, we present an overview of the Janus Personalized Web Anonymizer.

Figure 1 depicts an idealized configuration for Janus. A *Janus proxy* is located on each user's machine. This proxy realizes the Janus function and is at the same time responsible for filtering the (application) data stream from the user to a web-site. This includes HTTP messages and possibly cookies sent by the user's browser. A user's web-browser is configured to connect to the Janus proxy and to have all the communicating messages go through this proxy. Figure 1 depicts a situation with two users, Anne Miller and John Smith, and two web-sites, WALL STREET JOURNAL and NEW YORK TIMES. For concrete illustration, we elaborate next on an interaction between John Smith and NEW YORK TIMES.

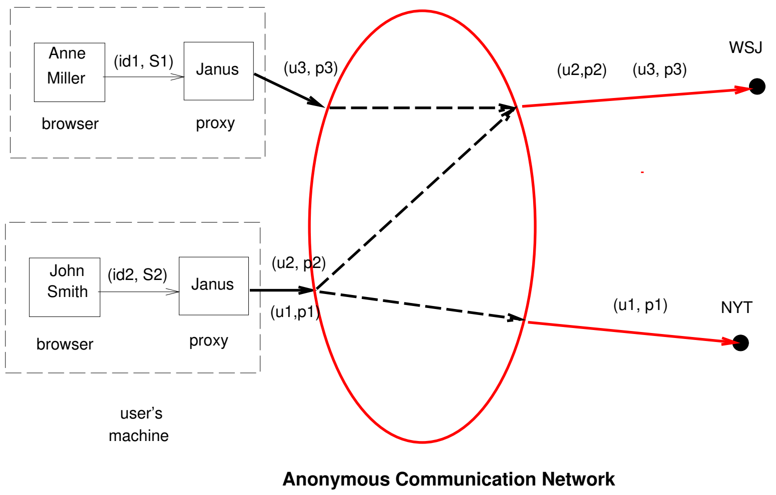


**Fig. 1.** Idealized Janus configuration

Assume that John's first request after starting up his web-browser is to connect to the NEW YORK TIMES web-site. This first request will be recognized by the Janus proxy, which replies by displaying its own HTML-document – a Janus authentication form – on John Smith's browser (see Fig. 2). This form asks for a username and a secret; it is recommended that the username be John's e-mail address, since this uniquely identifies John and can be used for e-mail functions as well.[2] John enters the requested information – $id2$ and $S2$, respectively – into the Janus form, and his browser sends it back to the Janus proxy (see Fig. 1). Janus receives this information and uses it for the rest of the session. From now on, and until the end of this session, Janus will only present John with HTML pages explicitly requested by him.

After Janus has obtained the required information about John Smith, it then forwards John's original HTTP-request for access to the NEW YORK TIMES site. As indicated in Sect. 1, John's browser possibly includes HTTP header fields and cookies in its messages, which reveal information about John. Janus filters such header fields and cookies from the request before forwarding the request to the web-site. If this is John Smith's first visit to the NEW YORK TIMES site, this site replies by sending a form to John asking for a username (subscriber id), a password and an e-mail address, in order to establish an account (see Fig. 3).

Now, instead of having to come up with a unique username and a secret password, John can simply fill in these fields with escape strings; e.g., "\U" for username, "\P" for password, and "\@" for e-mail address. These strings are recognized by Janus. Janus then takes John Smith's username ($id2$), secret ($S2$)

[2] John is free to select any secret, but he should provide the same secret at subsequent browsing sessions, so that Janus will generate consistent aliases.

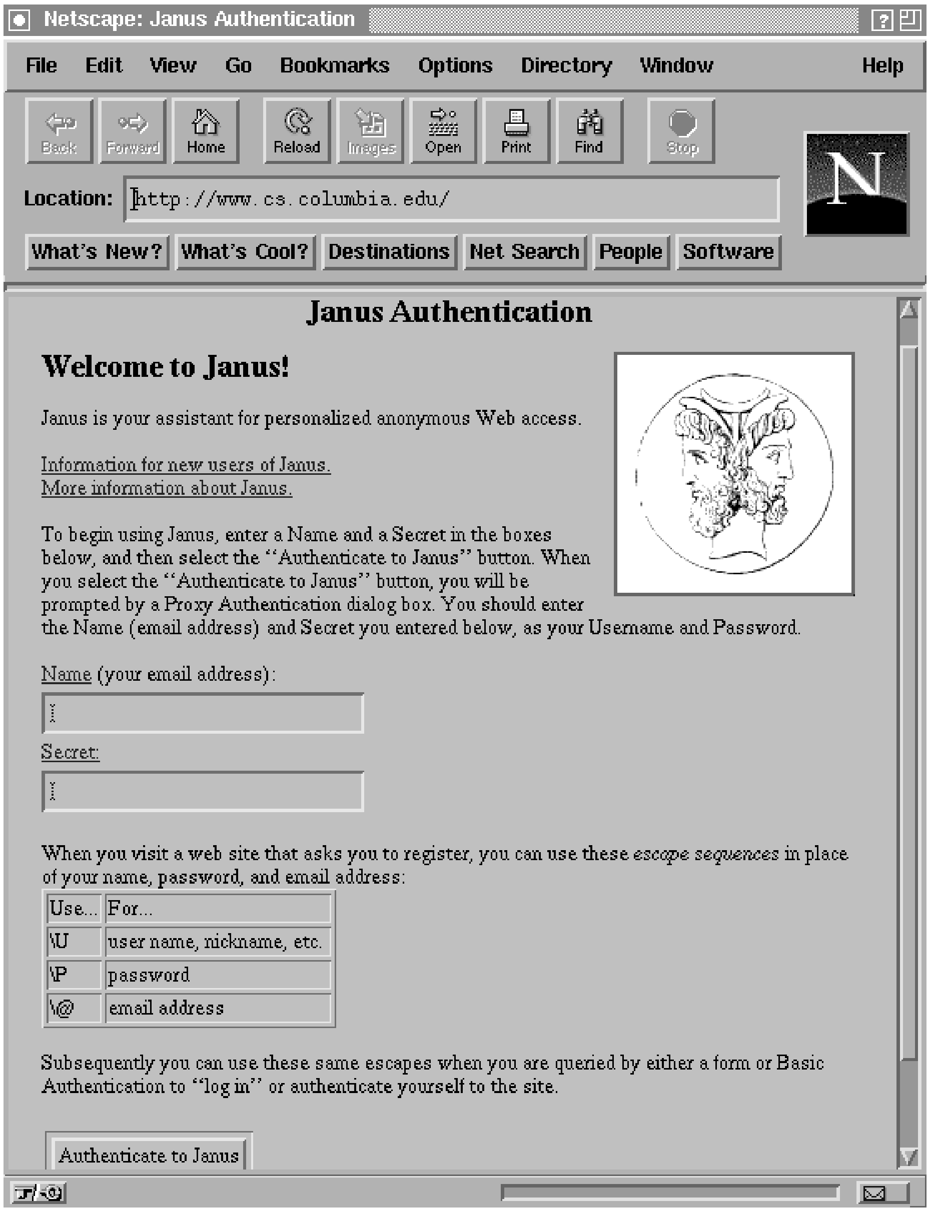


**Fig. 2.** Janus HTML page

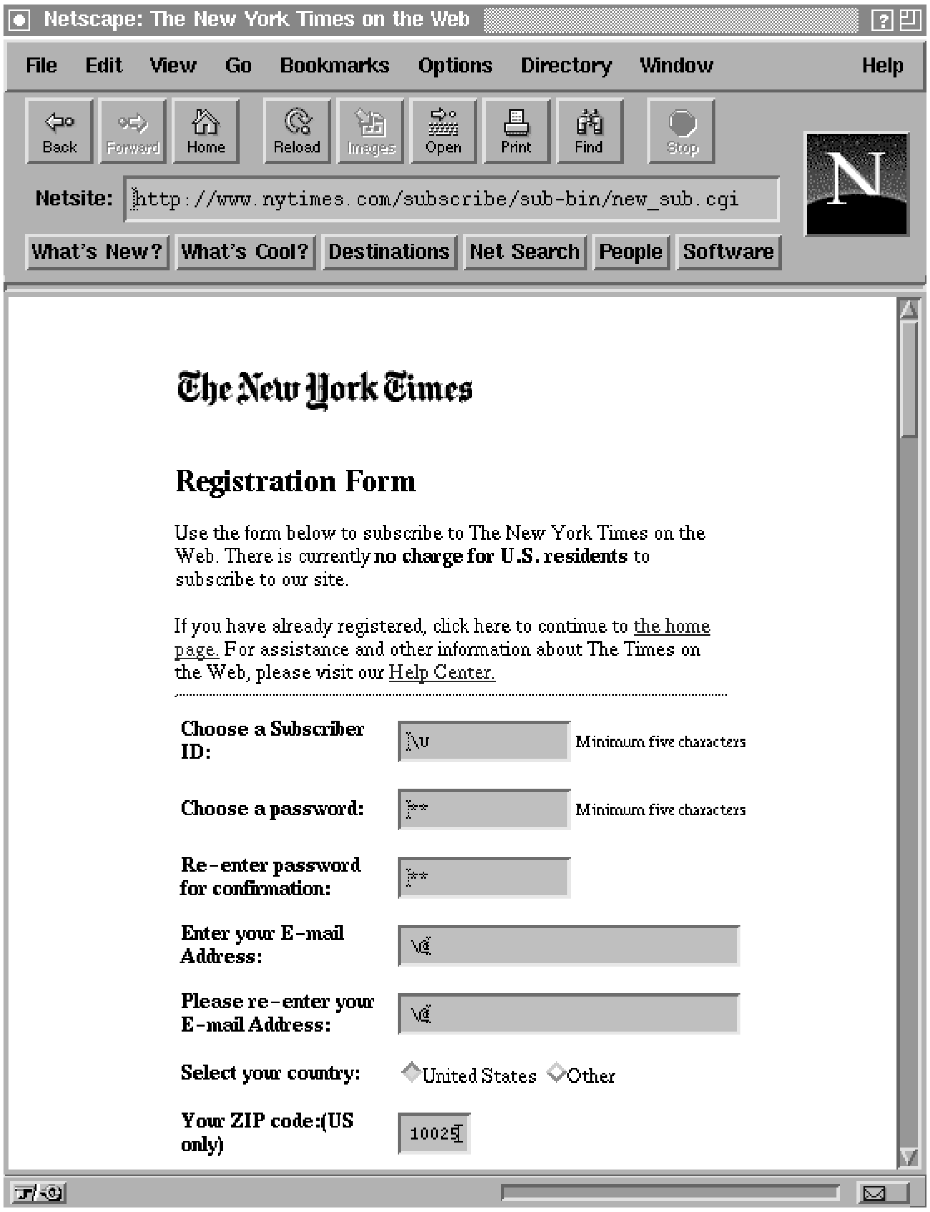


**Fig. 3.** New York Times HTML page (©1997 The New York Times)

and (part of) the domain-name of the NEW YORK TIMES site (`nytimes.com`) as input to compute an alias-username and an alias-password for the NEW YORK TIMES ($u1$ and $p1$ in Fig. 1) on John Smith's behalf. The computation of the alias-username and alias-password is done via the *Janus function* $\mathcal{J}$, described in Sect. 3. The issue of e-mail support by Janus is discussed in Sect. 4.

On subsequent visits of John to the NEW YORK TIMES site, this site will require that John authenticate himself, by submitting a username and a password. Upon sending the appropriate escape strings "\U" and "\P", Janus will automatically recompute $u1$ and $p1$ and reply by sending these values back to the NEW YORK TIMES site. John is freed from the burden of remembering the username and password of his NEW YORK TIMES account.

Since Janus already knows $id2$ and $S2$, only the second half of the above protocol is executed if John Smith decides to access additional web-sites during this browsing session. For example, if John accesses the WALL STREET JOURNAL site, the Janus function $\mathcal{J}$ is used to compute an alias-username, $u2$, and an alias-password, $p2$, as shown in Fig. 1. Distinct aliases are computed for John for each web-site, and these aliases are distinct from any generated for other users (e.g., in Fig. 1, the aliases $(u1, p1)$, $(u2, p2)$ and $(u3, p3)$ are all distinct).

The Janus system is directed towards providing anonymity of data content in web browsing. If the underlying network is allowed to trace connections, then Janus' utility might be in vain, since a web-site might be able to locate the Janus proxy on a user's machine (see Fig. 1). Thus, we suggest that the communication between a user and a web-site take place ideally over an *Anonymous Communication Network* (as depicted in Fig. 1). This kind of a network allows parties to send individual messages to each other and to reply to a received message anonymously. Simon in [S96] gives a precise definition for such an *Anonymous Exchange Protocol*. Research and implementation efforts for such anonymous networks are being carried out by several groups (e.g., [PW85, SGR97, GWB97]).

Figure 4 depicts a possible concrete realization of Janus. Here, a collection of users are located on a trusted Intranet behind a firewall. The Janus proxy is located on the firewall and all web browsing activity on the Internet is going through the Janus proxy. If the number of users behind the firewall is sufficiently large (e.g., a big corporate Intranet or an Internet Service Provider), then the fact that the web-sites can track a user back to this Intranet might be acceptable. A further refinement of this configuration is to allow a user to have a local Janus proxy on his/her machine, which in turn connects to the Janus proxy on the firewall. This refinement allows a user to keep his secret for Janus confined to his/her own machine and thus minimizes the trust placed on the Intranet.

Privacy of a data connection (e.g., a TCP connection) is an issue separate from anonymity. The connection between a Janus proxy and a web-site carries the user aliases. An eavesdropper on this connection can obtain a user's alias and then subsequently use it to access that particular web-site (impersonation). Note that this kind of scenario is even more problematic in today's web browsing without Janus, where the eavesdropper can obtain the real identity and possibly credit card information of the user. The use of SSL (see [SSL96]) can offer

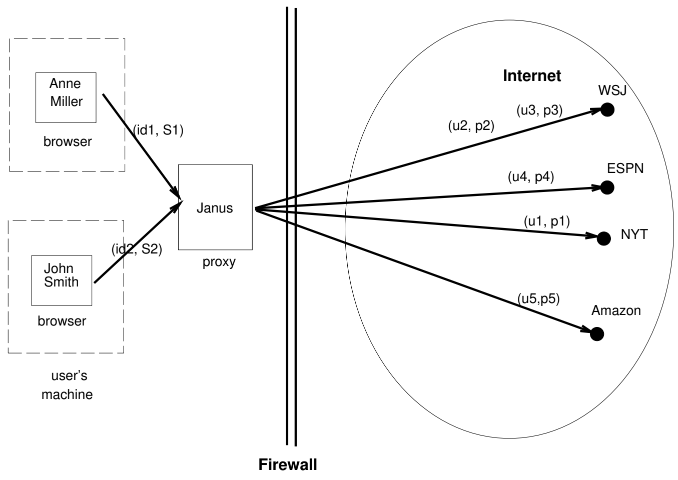


**Fig. 4.** Janus configuration for a large Intranet

protection if there is a direct connection between a user's browser and a web-site. But an increasing number of services have emerged in the form of proxies, where the proxy, rather than the browser, is supposed to be the endpoint of the secure connection (e.g., the SafePassage web proxy by c2.net, and the Janus proxy proposed in this paper). Developing a standard security tool that provides eavesdropper protection, like SSL, but that can be employed between a proxy and a web-site, unlike SSL, would be useful for such services, including Janus.

## 3 The Janus Function

In this section we discuss the requirements and construction of the Janus function. The construction is taken from [BGGMM97], where a formal specification and a proof that the construction meets these specifications are given. It is also demonstrated there that an attempt for a simpler construction is vulnerable to successful cryptanalytic attacks, and is hence not satisfactory.

**The Setting of the Janus-function.** As discussed in Sect. 2, a user inputs an identity $id$ and a secret $S$ at the beginning of a session. In the remainder of this paper, we assume that $id$ is the user's e-mail address. Whenever the Janus function is invoked, it relates to the current domain name of the displayed web-site (denoted by $w$), which is the third input to the Janus function and for which the resulting alias is to be used. The Janus function returns an alias, which is the pair $(u, p)$, i.e., the username and password of $(id, S)$ for web-site $w$.[3] Thus, $\mathcal{J}(id, w, S) = (u, p)$. We write $\mathcal{J}^u(id, w, S) = u$ and $\mathcal{J}^p(id, w, S) = p$.

[3] Construction of the alias e-mail address for the web-site is discussed in Sect. 4.

We assume that under normal operation a user $id$ does not reveal the secret $S$ to anybody, and that it reveals an alias-password $p$ only to web-site $w$. We assume, however, that each web-site maintains a list of registered usernames (i.e., users with accounts) at the site, and that this list may even be publicly available.

**Requirements.** The following are the requirements for the Janus function:

1. *Anonymity of users:* The identity of the user is kept secret; that is, a web-site, or a coalition of web-sites, cannot (except with negligible probability) determine the true identity of the user from his/her alias(es).
2. *Consistency:* For each web-site, each user is transparently provided with a consistent alias. The web-site can thus recognize the user-alias on repeat visits.
3. *Efficiently computable:* The function $\mathcal{J}$ should be efficiently computable given $id$, $S$, and $w$.
4. *Secrecy of passwords:* Each alias password remains secret at all times, except for the web-site for which it is submitted. In particular, an alias username does not give any information on any alias password.
5. *Uniqueness of aliases among users & impersonation resistance:* Given a user's identity and a web-site, a third party cannot find a different user identity which results in the same alias for that web-site. This also implies that the likelihood of two users having the same alias for the same web-site is negligible.
6. *Modular security & protection from creation of dossiers:* An alias of a user for one web-site does not reveal any information about an alias of the same user for another web-site. This also implies that a coalition of web-sites is unable to build a profile (dossier) based on the set of web-sites with which the user interacted.
7. *Single secret:* Given the user's (unique) identity and a single secret, the function $\mathcal{J}$ generates secure, distinct aliases as needed for each web-site. Note that this is a weaker assumption than that each user provides a *unique* secret. A user's secret is in the spirit of a regular login password, which does not have to be unique either.
8. *Acceptability:* The alias computed by the function $\mathcal{J}$ needs to be accepted by the web-site; e.g., it must have appropriate length and range.

During an interaction between a user and a web-site, there is at least one intermediary proxy involved (see Figs. 1 and 4). During a session of interaction, this proxy has private information about the user. A desired property is:

9. *No private user information stored in between interactions:* When a user is not interacting with a web-site, the Janus proxy does not maintain in memory any information that may compromise the above properties of the Janus function. (This property excludes, for instance, the simple approach of implementing a Janus function on the proxy by a look-up table.)

**Adversarial Setting.** We assume an active adversarial entity $E$. Suppose that $E$ learns either the secret $S$ or the alias-password of a user by breaking into the respective user-site or web-site, or by eavesdropping on the respective communication lines. In such case, the Janus function should maintain the above properties for all users about whom the adversary has not been able to learn their secret information. Furthermore, even for a user $id$ (and web-site $w$) for which the adversary can learn some site-specific secret information and relate it back to $id$, the user $id$ should still be able to remain anonymous and consistent in its interaction with other web-sites, as long as its secret $S$ is not revealed. The adversary $E$ can feed the Janus-function with chosen inputs (user identities, secret passwords, and web-site domain names) and examine the resulting aliases, i.e., $E$ can mount a *chosen plaintext* attack. $E$ can also consult arbitrary service-locations for registered alias-usernames. $E$ can perform any other polynomially-bounded computation.

**Design.** The design of the Janus function is based on pseudo-random functions and collision-resistant hash functions (see [GGM86] and [MOV97], respectively).

Let $h$ be a collision-resistant hash-function and let $f_\alpha$ be a pseudo-random function chosen from a pseudo-random function ensemble $F_l$ by using $\alpha$ as a seed. Let $||$ denote concatenation and $\otimes$ denote exclusive-or. In the following construction for the Janus function we assume for simplicity that $S = (S_1||S_2)$:

$$\begin{aligned} r_w^u &= f_{S_1}(w) \\ r_w^p &= f_{S_2}(w) \\ \mathcal{J}^u(id, w, S) &= h(r_w^u||(f_{S_1}(r_w^u) \otimes id)) \\ \mathcal{J}^p(id, w, S) &= h(r_w^p||(f_{S_1}(r_w^p) \otimes id)) \end{aligned}$$

In [BGGMM97], it is proven that the above construction meets the desired specification.

## 4 Supporting Anonymous E-mail by Janus

In this section, we show how the Janus system can be extended to allow web-sites to send e-mail back to users. We first note that Janus potentially can use the techniques or services provided by available anonymous remailers. For instance, a trusted intermediary $I$ can be added to the Janus system of Fig. 1, or such a role can be played by the firewall proxy in the configuration of Fig. 4. For a user with e-mail address $a$, Janus may compute an alias e-mail address $b$@hostname, where $b$ is an encryption of $a$ and "hostname" is the e-mail domain of the intermediary. Upon receipt of an e-mail message addressed to $b$@hostname, the intermediary will decrypt $b$ to obtain the user's e-mail address $a$, and forward the e-mail message to that address. The encryption and decryption functions are known both to the Janus proxy $J$ and to the intermediary $I$. This simple approach has the drawback that it violates the property of *modular security & protection* *from dossiers*, given in Sect. 3, because all web-sites see the same e-mail address for a user *id*. Furthermore, the intermediary $I$ has to be trusted with secret information that allows it to identify a user. However, in a configuration where Janus and the intermediary coincide, such as in Fig. 4, this last issue is not a concern.

In what follows, we describe an approach which does not violate the property of *modular security & protection from dossiers*. In [BGGMM97], an alternative approach is shown, which in addition does not require the intermediary to store secret, user-identifying information. To obtain this additional property, the e-mail is not forwarded to the user, but instead stored at the intermediary in anonymous mail-boxes until it is retrieved by the user.

**Design.** Our design is based on pseudo-random functions (see [GGM86]). Let $f_\alpha$ be a pseudo-random function chosen from a pseudo-random function ensemble $F_l$ by using $\alpha$ as a seed. Let $||$ denote concatenation and $\otimes$ denote exclusive-or. In the following construction of the e-mail address, let $K$ be a secret key, which is stored both on the intermediary $I$ and the Janus proxy (which coincide in the case of configuration of Fig. 4). Let $id, w, S$ be as in the construction of the Janus function of Sect. 3 and let $\kappa = (S||K)$.

When the user *id* enters the escape string "\@" into a form provided by a web-site $w$, Janus computes the alias e-mail address "$Email(id, w, \kappa)$@domain-name", where "domain-name" is the domain of the intermediary and "$Email(id, w, \kappa)$" is computed as follows:

$$\begin{aligned} r_w^u &= f_S(w) \\ Email(id, w, \kappa) &= r_w^u||(f_K(r_w^u) \otimes id) \end{aligned}$$

When intermediary $I$ receives e-mail addressed to $b$@domain-name, where $b = x||y$, it performs the following steps:

1. compute $z = f_K(x)$.
2. compute $id = z \otimes y$.
3. send the e-mail body to *id*.

Note that the constructions of $\mathcal{J}$ and *Email* are quite similar. Thus, given the proof in [BGGMM97] that $\mathcal{J}$ satisfies the properties given in Sect. 3, it is a simple exercise to show that adding $Email(id, w, \kappa)$@domain-name to the alias of a user *id* for web-site $w$ preserves all the properties of the alias username and password given in Sect. 3.

## 5 Payments

Some web-sites provide services which are not free. For instance, the WALL STREET JOURNAL and ESPN sites require the user to pay a monthly fee in order to access the news and information on their sites. The Janus system enables users to perform anonymous web browsing and to establish anonymous accounts. This browsing may lead to interactions with services that require payment. Once users have established communication with such services, they may face the problem of how to implement financial transactions without revealing their true identity. The problem of implementing anonymous financial transactions is independent of the basic functionality of the Janus system. Indeed, once a communication between a user and a web-site has been established, they may use any agreed-upon system that supports anonymous financial transactions. This consideration leads in a natural way to the idea of integrating Janus with a system like *JEPI* (Joint Electronic Payments Initiative) of the W3C [JEPI]; JEPI is planned to be a universal payment platform that will allow merchants and consumers to transact business over the Internet using many different forms of payment, possibly including anonymous payments. Furthermore, the research area of anonymous electronic cash and payment is very active; see, e.g., [CFN88, OO91, B93, FY93, LMP94, S96]. There are several systems available; see, e.g., DigiCash's Ecash [Ecash] and Cybercash [CC]. At present however, the support of anonymous payment by web-sites is somewhat limited, but it may grow as users become increasingly concerned about their privacy.

We briefly point out an alternative approach, in which a Janus system (e.g., as depicted in Fig. 4) may serve as a mediator for the purpose of anonymous payment. The nature of interaction between a user and the Janus proxy may enable the Janus system to assist in anonymous payments which are based on HTTP messages. In particular, the Janus system may pay a service $w$ a certain fee $x$ on the user's behalf, at the request of a user $id$. The user $id$ first pays Janus the requested amount $x$. When this transaction is completed (and the funds are at Janus' possession), Janus pays the service $w$ the requested amount. The transaction between the user and Janus may be anonymous, provided that Janus supports some means of anonymous payment. On the other hand, there is no need for the transaction between Janus and the service to be anonymous. Thus, Janus may enable anonymous payments, regardless of whether services are supporting such transactions.

The above sketch offers only a general scheme. There are several important issues that need to be addressed, both regarding implementation details, and regarding the legal ramifications of a service paying on behalf of a user, in particular in case of disputes. These issues may impose certain restrictions on the use of a Janus-based anonymous payment scheme, and are not addressed here.

## 6 Related Work

The work closest in spirit to our goal of anonymous personalized web browsing is the visionary paper of Chaum [C85] on *digital pseudonyms*. Chaum presented a general framework in which users maintain distinct pseudonyms for different organizations, such that pseudonyms cannot be traced or be related to each other. However, there is no concrete realization of pseudonyms with these properties that is similar to the one considered here. Chaum's concrete proposals are geared towards financial transactions and electronic payments. So for example, a digital pseudonym is created by having a user choose a 100-digit random number. This number is then (blinded and) signed by the user and verified for uniqueness and signed by a central entity (bank). Hence, our construction of aliases is orthogonal: they can potentially be used in lieu of the 100-digit random number. Our aliases are much shorter (as may be required by the *acceptability* property) and no central entity is needed to verify them in order to achieve the properties given in Sect. 3. Chaum's ideas are built upon in the NetBill project to achieve privacy in electronic transactions (see [CTS95]). An alternative is presented in [LMP94] for an anonymous credit card, in which a customer transfers funds via a trusted intermediary from his/her bank to the store's bank during a purchase.

There are several services currently operating that offer capabilities ranging from simple anonymous e-mail to sophisticated mixmaster remailers. See for example [PW85, GT96] for research papers on this subject and [C96] for an account on the *Mixmaster Remailer*. Such services focus on providing anonymity in e-mail exchanges, but either do not provide the *consistency* property or do not provide the *modular security & protection from dossiers* properties we require of Janus. Furthermore, remailers often use table-based translation and thus do not fulfill the *no private user information stored in between interactions* property. As a consequence, if they are broken into or if they are required by legal authorities to open their records (see for instance the Penet remailer incident, mentioned, e.g., in [GT96]), then anonymity is lost.

The *Anonymizer* (see [ANON]) is a service which provides an intermediate entity which filters HTTP headers for web browsing. While such capability provides anonymous web browsing, it does not provide a means to establish anonymous accounts or otherwise support anonymous personalized web browsing.

## 7 Conclusions

Commercial web-sites play an increasing role in the area of electronic commerce, which has a yet untapped potential. But in order to fulfill its promises, electronic commerce needs to offer tools which provide customers with easy, secure, anonymous yet personal web access to these sites. Such tools ultimately benefit as well the merchants running the web-sites: If users are assured that their privacy is protected, they will be more tolerant of the tracking and demographic data-gathering mechanisms that enable merchants to tailor their web pages in order to optimize sales. The Janus Personalized Web Anonymizer is a step in that direction.

### Acknowledgements

We thank Daniel Bleichenbacher, David M. Kristol, Peter Winkler and Moti Yung for many helpful comments. We also thank Udi Manber for pointing out the issue of privacy in Internet browsing, during an invited talk at SIGMOD'96.

## References


[ANON] The Anonymizer. http://www.anonymizer.com

[B93] S. Brands, Untraceable off-line cash in wallet with observer. *Crypto'93*, Springer-Verlag LNCS 773, pp. 302–318.

[BGGMM97] D. Bleichenbacher, E. Gabber, P. B. Gibbons, Y. Matias, A. Mayer, On personalized yet anonymous interaction. Technical report, Bell Laboratories, April 1997.

[CC] Cybercash. http://www.cybercash.com

[C85] D. Chaum, Security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM*, **28**(10), October 1985, pp. 1030–1044.

[C96] L. Cottrell, Mixmaster and remailer attacks. http://obsucra.com/ loki/remailer/remailer-essay.html.

[CFN88] D. Chaum, A. Fiat, M. Naor, Untraceable electronic cash. *Crypto'88*, Springer-Verlag LNCS 403, pp. 319–327.

[CTS95] B. Cox, J.D. Tygar, M. Sirbu, NetBill security and transaction protocol. *1st Usenix Workshop on Electronic Commerce*, July 1995.

[Ecash] An introduction to ecash. http://www.digicash.com/publish/ecash_intro/ecash_intro.html.

[FY93] M. Franklin, M. Yung, Secure and efficient off-line digital money. *20th ICALP*, Springer-Verlag LNCS 700, 1993, pp. 265–276.

[GGM86] O. Goldreich, S. Goldwasser, S. Micali, How to construct random functions. *J. of the ACM*, **33**(4), 1986, pp. 210–217.

[GT96] C. Gulcu, G. Tsudik, Mixing email with babel. *ISOC Symposium on Network and Distributed System Security*, 1996.

[GWB97] I. Goldberg, D. Wagner, E. Brewer, Privacy-enhancing technologies for the internet. *Compcon'97*.

[HTTP] R. T. Fielding, J. Gettys, J. Mogul, H. Frystik Nielsen, T. Berners-Lee, HTTP/1.1., Internet RFC 2068, 1996.

[JEPI] JEPI. www.w3.org/pub/WWW/Payments/

[LMP94] S. Low, N. Maxemchuk, S. Paul, Anonymous credit cards. *2nd ACM Conf. on Computer and Communications Security*, 1994, pp. 108–117.

[MOV97] A. Menezes, P. van Oorschot, S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997.

[OO91] T. Okamoto, K. Ohta, Universal electronic cash. *Crypto'91*, Springer-Verlag LNCS 576, pp. 324–337.

[PW85] A. Pfitzmann, M. Waidner, Networks without user observability – design options. *Eurocrypt'85*, Springer-Verlag LNCS 219, pp. 245–253.

[S96] D. Simon, Anonymous communication and anonymous cash. *Crypto'96*, Springer Verlag LNCS 1109, pp. 61–73.

[SGR97] P. Syverson, D. Goldschlag, M. Reed, Anonymous connections and onion routing. *IEEE Symposium on Security and Privacy*, 1997, to appear.

[SSL96] P. Karlton, A. Freier, P. Kocher, The SSL Protocol, 3.0. *Internet Draft*, March 1996.

[T96] D. Taylor, "The Webmaster: Web Site Memory with Cookies" *;login: (Usenix newsletter)*, October 1996, pp. 32–35.


This article was processed using the LaTeX macro package with LLNCS style